Towards Improving Data Validity of Cyber-Physical Systems through Path Redundancy

Zhiyuan Zheng and A. L. Narasimha Reddy
Department of Electrical & Computer Engineering
Texas A & M University
College Station, Texas 77843
{zhiyuanbj, reddy}@tamu.edu

ABSTRACT

Cyber-physical systems have shown to be susceptible to cyber-attacks. Incidents such as Stuxnet Attack and Ukraine power outage have shown that attackers are capable of penetrating into industrial control systems, compromising PLCs, and sending false commands to physical devices while reporting normal sensing values. Therefore, one of the critical needs of CPS is to ensure the validity of the sensor values. In this paper, we explore path diversity in SCADA networks and develop Path Redundancy to improve data validity. The proposed solution is shown to be able to effectively prevent data integrity attacks and detect false command attacks from a single compromised path or PLC. We provide detailed analysis on solution design and implement an application of the technique in building automation networks. Our cost-efficient and easy-to-deploy solution improves the resilience of SCADA networks.

Keywords

Cyber-Physical Systems; SCADA; Path Redundancy; BACnet; Building Automation Networks

1. INTRODUCTION

Cyber-Physical Systems (CPS) leverage communication technologies to monitor and control sensors and actuators in a physical system. The communication system used in CPS is called Supervisory Control and Data Acquisition System (SCADA). SCADA performs important functions in many national critical infrastructures such as power grid, oil/gas, water/sewage, and building automation. With the increasing need of remote and convenient control of distributed control devices and equipment, SCADA networks have started to incorporate Wide Area Network (WAN) and IP technologies to facilitate the Command and Control (C&C) process.

However, the incorporation comes at a cost: it creates new threats from potential cyber-attacks. Since SCADA controls many *safety-critical* sectors, its failure can cause irreparable damage to the physical system or jeopardize people who depend on it. Cyber security of CPS has attracted a tremendous amount of attention lately due to recent cyber-attacks that have been publicized in the news. To illustrate, in 2010, the Stuxnet worm sabotaged Iran's nuclear facilities through infecting Siemens S7 software and PLCs which control motor's spinning frequency [15, 23]. The attacker issued false commands from the PLC to periodically modify motor's spinning frequency while reporting normal sensing values back to the user. The recent Ukraine power outage [16] was the world's first power outage caused by hackers. The hackers compromised PLCs and leveraged them to issue unauthorized commands to open circuit breakers. The power outage impacted approximately $225,000$ customers over three distribution areas [16, 18].

Attacks toward CPS normally come from two sources: through *physical access* or *remote intrusion*. Recently, an increasing number of cyber-attacks infect CPS through remote intrusion where attackers gain remote access by infecting corporate network as the first step. Since SCADA networks are usually connected with corporate LANs, the adversary may first employ stealthy Malware or viruses to infect workstations or web servers, stealing VPN confidentials and gaining access to the control network.

Next, attackers may compromise devices and equipment in the control network to launch attacks. SCADA system uses distributed *Programmable Logic Controllers (PLCs)* and *Remote Terminal Units (RTUs)* to control and monitor different types of *Physical Devices (PDs)* such as sensors, valves, pumps, drives, boilers and generators. Typically, each PD is only connected with one PLC/RTU. If a PLC/RTU is compromised, the attackers could directly send malicious commands to the connected PDs, causing unexpected changes to the physical system. They could also deceive the SCADA Server with invalid sensing values, compromising the data integrity of CPS networks. Such two attacks, called *false command attacks* and *data integrity attacks* respectively, could lead to widespread, costly and hazardous damage to an industrial control system, as illustrated by Stuxnet [23]. Compared with the SCADA Server, PLCs and RTUs are more susceptible to malicious attacks: they are normally embedded devices with limited processing and storage capabilities and weak or no security protections such as authentication, encryption, and antivirus capabilities. Therefore, PLCs and RTUs could become *Single-Point-of-Failures (SPOFs)* of SCADA networks.

Some recent work [6, 10, 25, 27] has looked at redundancy to improve the control network resilience and to elim-

CPSS'17, April 02 2017, Abu Dhabi, United Arab Emirates
© 2017 ACM. ISBN 978-1-4503-4956-7/17/04...$15.00
DOI: http://dx.doi.org/10.1145/3055186.3055189

inate SPOFs. Peer-to-peer (P2P) overlay network was introduced to SCADA networks [10]. The paper [10] proposed a middleware-based approach to transmit packets between the SCADA Server and PLCs in a P2P network. Their approach is able to protect networks from node crashes and data integrity attacks that are located between the source and destination. However, they cannot detect/prevent attacks from a compromised PLC. Other works [6, 25, 27] suggest to employ redundant hardware components such as servers, substations, controllers, and buses. This requires significant changes to the current system architecture and the additional hardware cost makes the solution difficult to deploy.

Intrusion Detection Systems (IDS) [26, 28] and anomaly detectors [4, 19, 21] are effective in preventing and detecting malicious/malformed packets, but most work is focused on the Server-PLC channel. It is more difficult to monitor the PLC-PD traffic because: (1) most PLC-PD channels adopt serial links (*e.g.*, RS-485 and RS-232); and (2) the variety in PDs' vendors and applications affects traffic behavior. The lack-of-security in the PLC-PD channel makes PDs more vulnerable to cyber-attacks.

To improve the resilience and data validity of CPS, this paper explores communication path diversity (especially on the PLC-PD channel) and proposes Path Redundancy to allow redundant sensing paths between the Server and a PD. The redundant paths go through other existing PLCs and are only used for sensing (data acquisition) purposes. The redundant sensing values are used to validate the data collected from the original path. When multiple values or inconsistent values are reported on the same *Physical Object* through redundant paths, this may indicate either (1) ongoing data integrity attacks from a single path or (2) false command attacks have been conducted on the Physical Device. We consider cyber-attacks coming from three locations: (1) a *single* compromised PLC; (2) Man-in-the-Middle (MITM) attacks originating between the Server and a *single* PLC; and (3) attackers inside the network but not MITM. Regardless of attacker's location, Path Redundancy can **prevent** data integrity attacks and **detect** false command attacks on a PD. Our non-intrusive solution takes advantage of existing PLCs in the CPS so that it can be easily integrated into current deployments at minimal cost. Our solution is suitable for different types of CPS in terms of application environment and link types.

1.1 Contributions

The contributions of this paper are the following:

- We propose Path Redundancy to enable redundant data acquisition paths between the Server and a Physical Device, which prevents a single compromised PLC from becoming the SPOF of CPS;
- We provide a detailed study on the implementation and demonstration of our solution in building automation networks;
- Based on our attacker model, our solution can effectively prevent ongoing data integrity attacks from a single path and detect false command attacks toward a Physical Device;
- Our solution is cost-efficient (in both computational and deployment cost), flexible, easy-to-integrate with existing deployments, and suitable for different types of CPS.

1.2 Paper Structure

The paper is structured as follows. In Section 2, we provide an overview of a typical SCADA system architecture and communication patterns. We also describe attacker models and the design requirements. We present the design of Path Redundancy in Section 3. Section 4 provides a detailed study on an application of our technique in building automation networks. The evaluation results are described in Section 5. Related work is presented in Section 6 and the conclusion as well as future work are provided in Section 7.

2. PRELIMINARIES

2.1 SCADA Architecture

A generalized SCADA architecture is presented in Fig. 1. SCADA adopts a three-level hierarchical structure and usually connects with the corporate LAN. It contains four main components located at three levels: *Supervisory Level*, *Automation Level*, and *Field Level*, as described below.

Sensors and Actuators are directly interfaced to the plant or equipment. They are responsible for sensing physical conditions and make changes to these conditions. Sensors convert different physical parameters (*e.g.* temperature values, current and pressure) into electrical signals (analog or digital). Actuators are used to manipulate certain equipment such as water pump, valves and electrical relays. They are also called *Physical Devices (PDs)*.

PLCs and RTUs are geographically distributed in the CPS to conduct direct control and monitor of the physical plant. Typically, sensors and actuators are attached to a designated PLC or RTU, either wired (*e.g.*, through serial/Ethernet interface) or wireless (*e.g.*, through ZigBee [2]). PLCs and RTUs are basically "field control devices" that execute automation tasks based on inputs from sensors, control logic, or commands from the SCADA Server.

PLCs and RTUs perform similar functions. However, RTUs are more suitable for gathering telemetry data over large geographical areas because they normally use wireless communication; whereas PLCs are more suitable for local control, like assembly lines in factories because they are designed with multiple inputs and outputs. In this paper, we use PLC to represent both field level controllers.

SCADA Server, or also called *HMI (Human-Machine Interface)* or *operator workstation*, interacts with PLCs and RTUs through IP encapsulated SCADA protocols to monitor the physical process and issue control commands either autonomously or by operator interaction. The purpose of the Server is to acquire data from the field, store data points for historical purposes, present operators with graphical display of the state of the process, provide a channel for operators to manipulate the physical devices, and issue alarms when events happen.

Communication Channel is the medium for data transmission from one to another. SCADA contains two types of communication channels: the *Server-PLC channel* and the *PLC-PD channel*. The Server-PLC channel usually adopts IP encapsulated SCADA protocols based on various forms: wired (*e.g.*, through fiber optics or copper cable) or wireless (*e.g.*, through radio or satellite communication). Popular SCADA protocols include but not limited to DNP3 [1], Mod-

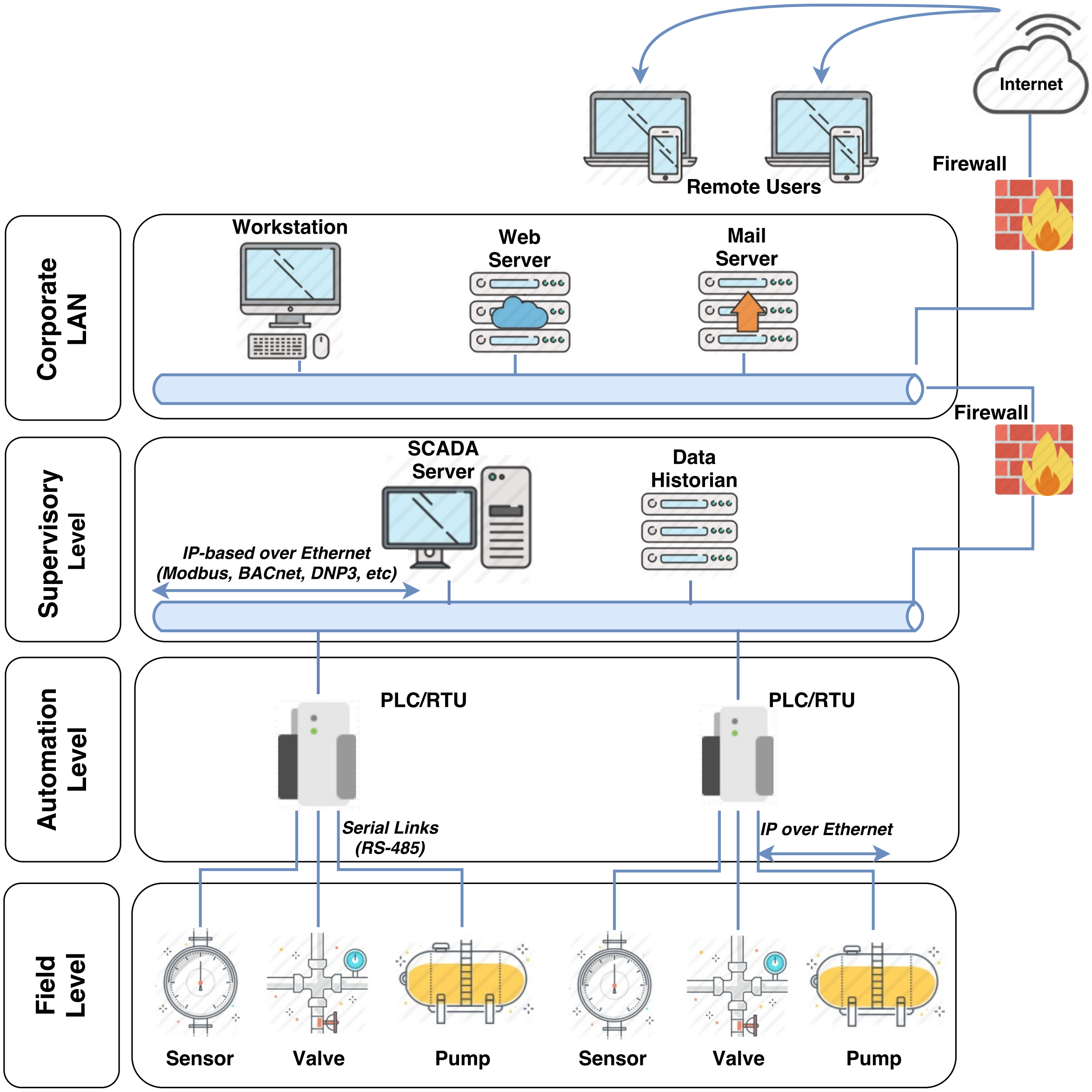


Figure 1: Architecture of a typical SCADA Network.

bus [17], and BACnet [3]. The PLC-PD channel normally uses two types of media: (1) serial interface (*e.g.*, RS-485, RS-232, and RS-422) [12] with standard SCADA protocols or vendor's proprietary protocols; and (2) IP-based SCADA protocols over Ethernet.

2.2 Attacker Model

Attackers who have gained access to the control networks may launch attacks from different locations, depending on their penetration capabilities. Fig. 2 presents the attacker model considered in this paper. We assume the SCADA server is equipped with adequate security protections and thus safe from cyber-attacks. We consider attacks from *three locations*: (1) a single compromised PLC; (2) MITM (Man-in-the-Middle) attacks originating between the Server and a PLC; and (3) attackers inside the network (not MITM).

A single compromised PLC. Attackers may infect a PLC using different approaches such as a USB flash drive and phishing emails containing Malware or viruses. Once the PLC gets infected, the attackers could take over the PLC by modifying its firmware or control logic, and use it to launch attacks. In this paper, we assume the adversary is only capable of compromising one PLC. Protection against multiple compromised PLCs is not considered in this paper.

MITM (between the Server and a PLC). This attack can be launched through several approaches: attackers may (1) infect a bump-in-the-wire device such as the network firewall or a switch between the two communication hosts, or (2) use ARP messages to trick hosts in the network to send their packets through the attacking host. Through either approach, the attackers are able to freely observe, intercept and manipulate unencrypted SCADA packets in transit. We assume the communication path between the Server and a *single* PLC can be compromised, while other PLCs are not affected.

Attackers inside the Network (not MITM). This

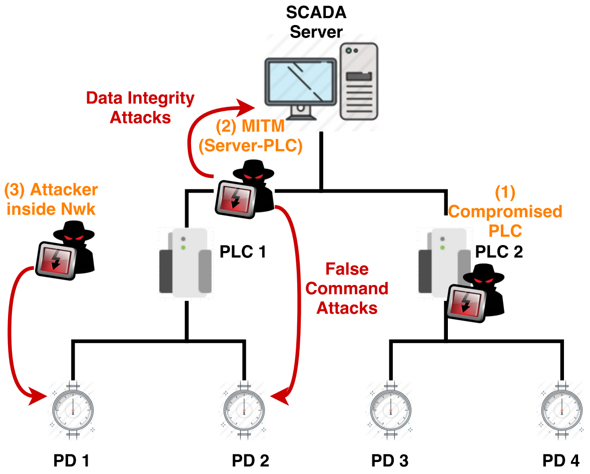


Figure 2: Attacker models considered in the paper.

describes attackers who are able to launch stealthy attacks from somewhere in the network, but not MITM. In this case, attackers are able to send packets to either the Server or a PLC (by spoofing itself as a legitimate device), but are unable to observe/intercept packets in transit in the Server-PLC channel.

Regardless of the location of attackers, without sufficient security features, attackers are able to launch **false command attacks** and **data integrity attacks**. False command attacks refer to malicious control commands that are issued to a PD (usually an actuator or a drive) trying to cause dangerous consequences to the control system. Data integrity attacks describe false or untruly sensing values reported to the SCADA server on behalf of a PLC. Such attacks aim to deceive the Server with invalid system operational information. In particular, some attackers may combine replay attacks with normal sensing values to hide their ongoing false command attacks; while others may cause the Server to issue false control commands based on the false sensing data and the control loop. Therefore, our attacker model consists of six attack scenarios (two attacks at three locations).

2.3 Design Requirements

The goal of this paper is to develop countermeasures against these types of attacks, improving the resilience and the data validity of cyber-physical systems. We are interested in a solution that can meet the following requirements:

- **Effectiveness.** The solution should be able to effectively thwart or detect different attacks with good accuracy. In should improve system resilience without introducing new threats or vulnerabilities;
- **Cost-efficiency.** Minimal computational and deployment cost is expected from both hardware and software aspects;
- **Practicality.** The solution should be able to be implemented and flexibly used in real-world CPS. Computational and storage capabilities of embedded devices, communication bandwidth, QoS (Quality-of-Service), and latency issues should be handled properly;
- **Easy-to-Deploy.** Since most SCADA networks have already been deployed or currently under deployment, our solution is expected to be easily integrated with current SCADA deployments with minimal change.

3. PATH REDUNDANCY DESIGN

To improve the data validity of SCADA networks, we propose the Path Redundancy which enables redundant sensing paths between the Server and each PD. Fig. 3 illustrates the idea of our solution. For example, PD#2 has a unique communication path with the Server (Server–PLC#1–PD#2) used for both sensing and actuating purposes. In addition to the existing path, we enable an additional path as Server–PLC#2–PD#2 (the dashed orange lines in Fig. 3). This redundant path is *ONLY* used for sensing purposes, NOT for actuating. The redundant path goes through PLC#2 which is used to control and monitor PD#3 and PD#4. Similarly, PLC#1 is now responsible for sensing data points stored on PD#3 and PD#4 (the dashed red lines in Fig. 3).

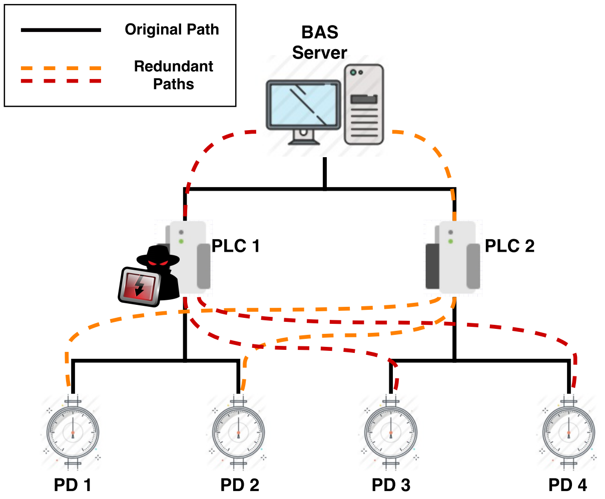


Figure 3: Redundant communication paths are enabled for sensing purpose.

Such change only affects data acquisition packets from the Server. All the other packets, including control and actuating packets, remain unchanged. Whenever the Server wants to acquire an object value on PD#2, it generates two sensing request packets to two different PLCs (one goes to PLC#1, and the other goes to PLC#2). The SCADA Objects used in both request packets are different, but they both reflect to the same Physical Object on PD#2. Both PLCs forward/reformat the request to PD#2 who responds with two independent packets to the Server through two paths. The Server receives two responses and uses the redundant data to validate the original. If the two values on the same Physical Object are inconsistent or unexpected, it indicates that one PLC or communication path is compromised (since we assume only one PLC/path can be compromised). The Server can take further investigation to detect and isolate the compromised component.

The above example illustrates the basic idea of the **Path Redundancy**. While most of the description in this paper describes only two paths, this idea can be generalized to more than two paths and quorum like policies or majority voting can be carried out to mask over incorrect values when more than two paths are available. Specific policies are determined based on different redundancy levels and constraints of the system. Typically, there are two main constraints: (1) limited processing and storage capabilities of PLCs, and (2) limited distance and node capacities of serial interface in the PLC-PD channel. Hence, we consider variety in two parts: *Number of Redundant Paths* and *Number of Data Objects.*

Number of Redundant Paths. Based on specific application environment, vulnerability risks and redundancy levels, multiple redundant paths can be enabled to further strengthen the security protection and increase the confidence of data validity. Multiple redundant paths solution allows the system to automatically mask out the incorrect values and immediately identify the compromised path/PLC. This can greatly shrink incident response time and minimize potential damage impact on the physical system.

Number of Redundant Data Objects. Since our solution suggests the use of existing PLCs to harden system security protection, adequate computing and storage capabilities are required at the Server and PLCs. Specifically, the support of multiple data replicas on the same Physical Object is required at the Server, and a PLC should be able to maintain data objects from the extended PDs. However, some PLCs or Server may not have enough storage or computational capability to support all the objects in the system. In this case, dynamic policies can be adopted which chooses a subset of Physical Objects for Path Redundancy. The subset of Physical Objects can be chosen based on their levels of criticalness and vulnerabilities. Many SCADA networks are capable of supporting the Path Redundancy with existing PLCs. Section 4 provides a detailed study of the situation in building automation networks.

For Path Redundancy with one redundant path, the PLC which is responsible for both sensing and actuating purposes toward a Physical Device is called the **Primary PLC** of that Physical Device, and the communication path passing through the Primary PLC is called the **Primary Path**. The redundant PLC that is only used for data acquisition purpose is called the **Secondary PLC** of a Physical Device, and the **Secondary Path** refers to the communication path passing through the Secondary PLC.

The reason to avoid control packets in Path Redundancy is three-fold. First, the control function of a PLC towards extended PDs increases the potential degree of impact when a PLC is compromised. In this case, it could be even more difficult to control and restore the system. Second, the ACKs of control requests normally contain insufficient information for the source to validate data validity. Finally, when duplicate control commands arrive at PDs, the additional buffering and cross validation of commands may be needed close to the PDs, making deployment potentially more difficult.

4. CASE STUDY: BUILDING AUTOMATION NETWORKS

This Section illustrates the design and implementation of Path Redundancy in building automation networks. First, we introduce building automation networks and the current data acquisition method. Next, we describe implementation challenges. Finally we describe our modifications in both hardware level and software level.

4.1 Background of BAS

Building Automation System (BAS) is a distributed control system responsible for the heating, ventilation, air conditioning (HVAC), lighting, security, fire and access control of a “smart building”. The main functions of a BAS include maintaining the efficient operation of a building, reducing energy consumption and operational cost, keeping temperature within a specified range, and providing lighting and access control.

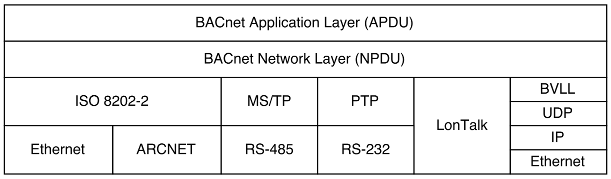


Figure 4: The layered architecture of BACnet protocol.

BACnet$^{\mathrm{TM}}$ is the standard data communication protocol for building automation and control networks. It was developed by ASHRAE [3] to standardize communications between building automation devices from different manufacturers, allowing information to be exchanged and equipment to work together easily. BACnet adopts a layered protocol architecture based on the collapsed version of the OSI model, as shown in Fig. 4. BACnet application layer and network layer are used with different media types including MS/TP, LonTalk, ARCNET, and Ethernet. BACnet/IP (based on UDP/IP) is widely used because it leverages IP internetworks to effectively connect geographically scattered equipment together.

To facilitate data communication, BACnet defines a collection of abstract, network-visible view of data structures called **Objects**. BACnet objects offer virtual data structures that enable identifying and accessing information without revealing internal designs of each device. It defines 54 types of objects including but not limited to *Analog/Binary Input*, *Analog/Binary Output*, *Schedule*, *Alert*, and *Device.* The BACnet standard does not require all the objects to be present in every BACnet device. The choice of which objects are present in a BACnet device is determined by the device’s application needs. *Device* is a special type of object that must be present in every BACnet device. Each BACnet device is assigned with a *Device ID* which is unique in BACnet internetworks.

Services are functions by which a BACnet device manipulates or accesses properties of BACnet objects. For example, services such as *ReadProperty* and *WriteProperty* are used for data sharing, and services like *WHOIS*, *IAM*, *WHOHAS*, *IHAVE* are used for device and object discovery. BACnet services are classified as *confirmed* or *unconfirmed.* Confirmed services adopt request-acknowledgement state machine while unconfirmed services do not expect an ACK. Devices respond to an unconfirmed service (*e.g.*, WHOIS) with another unconfirmed service (*e.g.*, IAM). Table 1 shows a list of common BACnet services.

In a BACnet/IP network, the BAS Server communicates with multiple **Field Panels (FPs)** (also called PLCs) that are scattered in the network. Multiple BACnet Field Panels that are located close to each other form a subnet which is called a *Building Level Network (BLN).* All BLNs in a BAS form the BACnet internetworks. Each Field Panel connects

Abbr	Service	Type	Description
CAA	ConfirmedAcknowledgeAlarm	Confirmed	Tell the sender of an alarm that the alarm has been received
CCOV	ConfirmedCOVNotification	Confirmed	Tell the subscriber that a COV has happened in an object
CEN	ConfirmedEventNotification	Confirmed	Tell another device of an error or fault has occured
CPT	ConfirmedPrivateTransfer	Confirmed	Sends a vendor-proprietary message to another device
IAM	I-AM	Unconfirmed	Affirmative response to WHOIS, broadcast
IHAVE	I-HAVE	Unconfirmed	Affirmative response to WHOHAS, broadcast
RP	ReadProperty	Confirmed	Reads the value of a particular object property
RPM	ReadPropertyMultiple	Confirmed	Reads the values of multiple object properties
RR	ReadRange	Confirmed	Reads trend-longs of a BACnet device
SCOV	SubscribeCOV	Confirmed	Sent by a device to request that it be told of COVs in an object
TS	TimeSynchronization	Unconfirmed	Notify the device of the correct current time
WHOHAS	WHO-HAS	Unconfirmed	Ask which BACnet device holds a particular object, broadcast
WHOIS	WHO-IS	Unconfirmed	Ask about the presence of a particular object, broadcast
WP	WriteProperty	Confirmed	Writes a value to an object property
WPM	WritePropertyMultiple	Confirmed	Writes multiple values to different object properties

Table 1: BACnet service abbreviations and descriptions.

to several Physical Devices which form a local *Field Level Network (FLN)*.

In today's BAS deployments, BACnet/IP is widely used as the communication protocol in the Server-FP channel. BACnet message types include *unicast*, *broadcast*, and *multicast*. Since broadcast messages are not allowed to pass across IP routers, each BLN employs a special Field Panel called *BACnet/IP Broadcast Management Device (BBMD)* to handle broadcast messages. The BBMD in a BLN repackages the message as a unicast and transmits it to its peers in other BLNs who broadcast the message on their local BLNs. BBMD can be either a physically distinct device or integrated into a BACnet FP.

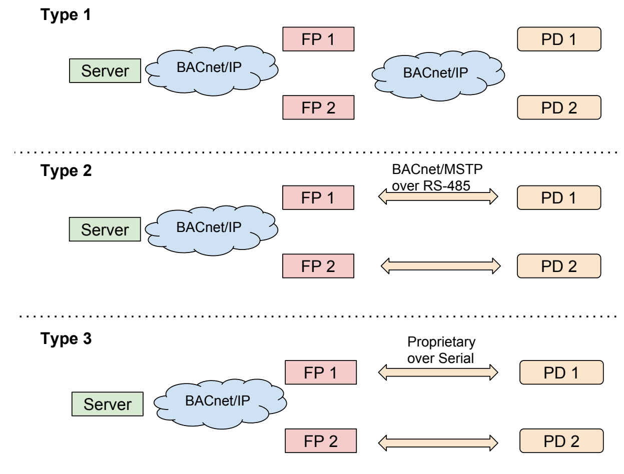


Figure 5: Three types of common FP-PD communication channels in current BACnet networks.

Three types of channels are found in the FP-PD communications, as shown in Fig. 5. These types are (1) BACnet/MSTP over serial links (*i.e.*, RS-485 / RS-232), (2) proprietary protocols or vendor-specific protocols (*e.g.* Siemens P1 Protocol) over serial links, and (3) BACnet/IP over Ethernet. The majority deployments adopt serial interface RS-485 in the FP-PD channel, and BACnet/MSTP is preferred over vendor's proprietary protocols because the use of proprietary protocols restricts the owner's flexibility and cost of replacement, service and switching to other vendors.

RS-485 [22] supports local networks and multidrop communication links. It can support up to 32 devices (nodes) spanning within $1,200$ m ($4,000$ ft). More devices can be connected using repeaters, up to the addressability limit (usually 256) of the devices used [8]. **Ethernet** over copper wiring supports up to 100 m (328 ft) whereas Ethernet over Fiber-optic can achieve a maximum distance of $80-100$ km ($50-62$ mi). The main drawback of RS-485 is that it only offers data transmission speed of at most 35 Mbps, whereas Ethernet can support the speed of $100-1,000$ Mbps.

When BACnet/MSTP or BACnet/IP is used in the FP-PD communication channel, the Field Panel acts like a BACnet Router and local controller; however, when proprietary protocols are used in the FP-PD channel, the Field Panel acts like a BACnet Gateway and local controller. In this case, there are two approaches to design these kinds of gateways:

(1). All of the physical devices' data points are aggregated into one "super BACnet device" that usually has a very large collection of BACnet objects.

(2). The Field Panel acts like a BACnet router to a "virtual network" of BACnet devices, and emulates the BACnet objects in each of those devices. In this case, there would be multiple distinct BACnet Devices in the gateway.

4.2 Current Data Acquisition Approaches

Data acquisition in a BAS usually takes two forms: (1) *Server-initiated* requests to retrieve data objects stored on a PLC; and (2) *FP-initiated* unsolicited response. Server-initiated method is usually used more frequently than the other. In the first method, the BAS Server usually sends ReadProperty or ReadPropertyMultiple services to a Field Panel according to different timers or commands from the system operator. The FP-initiated method, however, is usually used by a Field Panel to spontaneously report a Change-of-Value (COV) event. When a subscribed object value changes beyond the specified threshold, the Field Panel will send a ConfirmedCOV (CCOV) request to the Server to report the COV event. CCOV messages are limited to the Server-subscribed objects and are considered as a substitute for Server-initiated data acquisition messages.

4.2.1 Hardware Level

If Ethernet is used as the communication interface, network switches are responsible for routing Ethernet packets. RS-485 communications normally require a hard wired, point-to-point cable connection between devices. It is designed for a master/slave topology where the master device polls each slave, wait for the response, and move to the next one. This architecture avoids data collision at the hardware level.

4.2.2 *Software Level*

If proprietary protocols are used in the FP-PD channel, the Field Panel caches object values of its connected Physical Devices in the local **Object Database**; whereas if the Field Panel acts as a BACnet router between BACnet/IP and BACnet/MSTP, it does not cache values, just forwards messages.

If a BACnet Field Panel maintains the local Object Database, it periodically updates the database by sending messages to its connected Physical Devices. When a Field Panel receives a BACnet read request, it is up to the vendor's application to determine how to prepare the response data. Two main types of mechanisms are found in BACnet Field Panels: **Response-with-Update (RWU)** and **Response-without-Update (RWOU)**.

Whenever a data acquisition request arrives at a Field Panel, the Field Panel of Type RWOU will directly respond to the request based on its local Object Database; whereas Field Panels of Type RWU will need to update the target content first before sending the response. Compared with RWU, RWOU minimizes the Round-Trip Time (RTT), but it may send stale data back to the Server, if the updating frequency of Object Database is low. It is up to the vendor's decision in choosing the design strategy, and the RWOU is relatively more widely used.

In addition, the vendor's software maintains an **Object Name Table (ONT)** that projects different Physical Objects to BACnet address pairs: <BACnet FP, BACnet Object ID>. The ONT is stored at all kinds of BACnet devices, including the BACnet Server, Field Panels, BBMD devices, and BACnet Physical Devices. Table 2 illustrates a sample ONT stored at the BAS Server. For example, "FL1_RM2_TEMP_SETPOINT" is mapped to < FP#1, AnalogValue#1 >. Since each Physical Device is only controlled and managed by one Field Panel, this is a one-to-one mapping. Similarly, the ONT at a BACnet Field Panel/BBMD maps BACnet Objects to Physical Objects on its connected Physical Devices.

Physical Object Name	**BACnet FP**	**BACnet Object**
FL1_RM2_TEMP_SETPOINT	FP 1	AnalogValue 1
FL2_CHL_WATER_PRESSURE	FP 2	AnalogValue 2

Table 2: A sample Object Name Table stored at the BAS Server.

Therefore, if the system operator wants to acquire the current value of "FL1_RM2_TEMP_SETPOINT", the application at the BAS Server checks the ONT, interprets the Physical Object to the BACnet Object address pair (FP#1 : AnalogValue#1), constructs the ReadProperty request with the BACnet Object, and sends the packet to the target Field Panel. Configurations and maintenance of the ONT is not part of BACnet Standard. Some vendors manually configure the ONT on each BACnet device during the initial setup period.

4.3 Challenges

In order to apply Path Redundancy to a BAS, we need to solve the following three challenges.

Hardware Level. The FP-PD communication channel can take two types of media (serial interface v.s. Ethernet) and two protocols (BACnet v.s. proprietary protocols). We need to develop hardware support for Path Redundancy in all cases, particularly for channels using serial interface.

Software Level. The BACnet devices employ Object Name Table for object naming translation. Current naming translation is a one-to-one mapping between Physical Objects and BACnet Objects. Path Redundancy requires a one-to-multiple ONT mapping.

Other Limitations. There are some other practical limitations that we need to take into account. For example, the capacity and distance constraints of RS-485 may limit the use of existing FPs for Path Redundancy.

4.4 Hardware Level Modifications

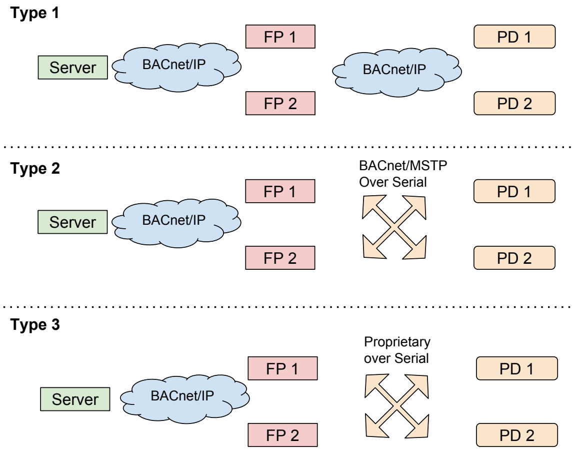


Figure 6: Hardware level changes for Path Redundancy.

The hardware level modifications consider both Ethernet and serial links used in the FP-PD communication channel. Fig. 6 shows the changes for all three types of FP-PD channels. If BACnet/IP over Ethernet is used, no hardware modifications are needed. The Ethernet switch is responsible for routing packets to the target Physical Device. If BACnet/MSTP or proprietary protocol is used over serial links (RS-485 in this case), we need serial hubs/splitters. For example, if one redundant path is applied in a network: two FPs sensing values on a PD (cf. FP#1 and FP#2 in Fig. 6), we can connect a one-to-two serial hub to each FP and a two-to-one serial hub in front of each PD. In this case, packets from any FP can be directed to both PDs.

4.5 Software Level Modifications

In software part, we modify the Object Name Table stored at the Server and all BACnet Devices. Table 3 demonstrates the changes for the ONT at the BAS server when one redundant path is applied. For each Physical Object, the ONT includes two BACnet address pairs: <BACnet FP, BACnet Object ID>, one corresponds to the primary path, and the other corresponds to the secondary path. For instance, the "FL1_RM1_TEMP_SETPOINT" maps to <FP#1, AnalogValue#1> (the primary path) as well as

<FP#2, AnalogValue#11> (the secondary path). The specific BACnet Object IDs on the redundant Field Panel can be any IDs that are not in use. When the Server wants to send a ReadProperty or a ReadPropertyMultiple request, the application interprets the Physical Object into two BACnet address pairs, constructs two packets, and further sends out to two FPs. Similarly, the ONT at a FP/BBMD is also changed in the same way. Different vendors can use their own methods to maintain and update the ONT.

Physical Object Name	BACnet FP	BACnet Object
FL1_RM1_TEMP_SETPOINT	FP 1 (Primary) FP 2 (Secondary)	AnalogValue 1 AnalogValue 11
FL2_CHL_WATER_PRESSURE	FP 2 (Primary) FP 1 (Secondary)	AnalogValue 2 AnalogValue 22

Table 3: Modified ONT for Path Redundancy.

4.6 Practical Limitations

In a typical BAS network, each building (or a BLN) contains multiple FPs and a BBMD device. These FPs are located close to each other and their controlled Physical Devices. When necessary, RS-485 repeaters can be used to extend the distance and node (device) capacity.

Overall, our solution is applicable to all types of BAS FP-PD channels and it can be flexibly adapted to CPS with different kinds of constraints.

5. EVALUATION

To evaluate the effectiveness and accuracy of our solution, we implemented Path Redundancy in an emulated Building Automation System using Raspberry Pi devices and a PC-based control server. We are interested to see if our solution can effectively protect the system from six attack scenarios described in Section 2.2 (two attacks launched from three locations).

5.1 Emulation Settings

Our testbed implements the Path Redundancy with one redundant path on all the objects. The testbed contains five components: the BACnet Server, a Primary FP, a Secondary FP, a BACnet PD, and an attacker. The attacker is a stand-alone device under the *MITM* and *Attacker inside the Network* case, but is mounted on a FP under the *Single Compromised FP* case. Fig. 7 shows the topology of the BAS testbed used in the paper. The Server is emulated by a Ubuntu 16.04 system running on Intel Core i5-2320 3.00GHz CPU with 8G RAM. FPs are emulated by two Raspberry Pi 3 Model B devices, the PD is emulated by a Raspberry Pi Model B+, and the attacker is emulated by a Raspberry Pi 2 Model B. We assume BACnet/IP is used in both Server-FP and FP-PD communications. All the devices are connected using Ethernet within the same subnet, and the bandwidth is 1 Gbps.

We modified the OpenSource BACnet Stack [13] to enable the FPs (here as BACnet Routers) forwarding packets between the BACnet Server and the PD. For BACnet/IP networks, no hardware modifications were needed, and about 800 lines of code were modified/added for the software modifications. Thus, our solution is easy-to-deploy with small modifications. After the changes, ReadProperty and ReadPropertyMultiple requests from the Server to the PD are routed through both FPs, while all other requests including WriteProperty and WritePropertyMultiple are only handled by the Primary FP.

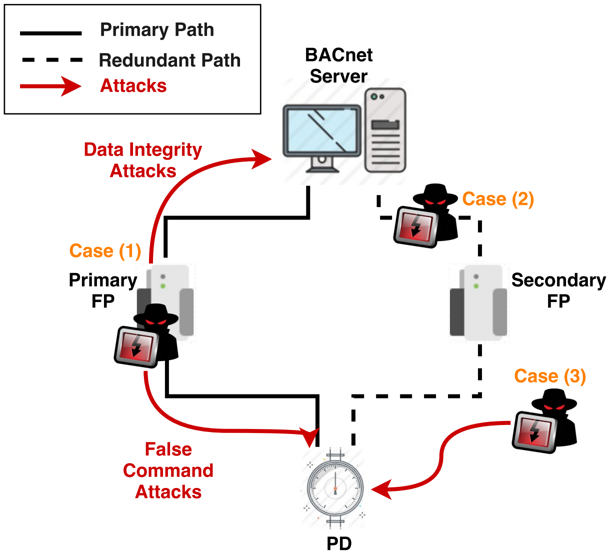


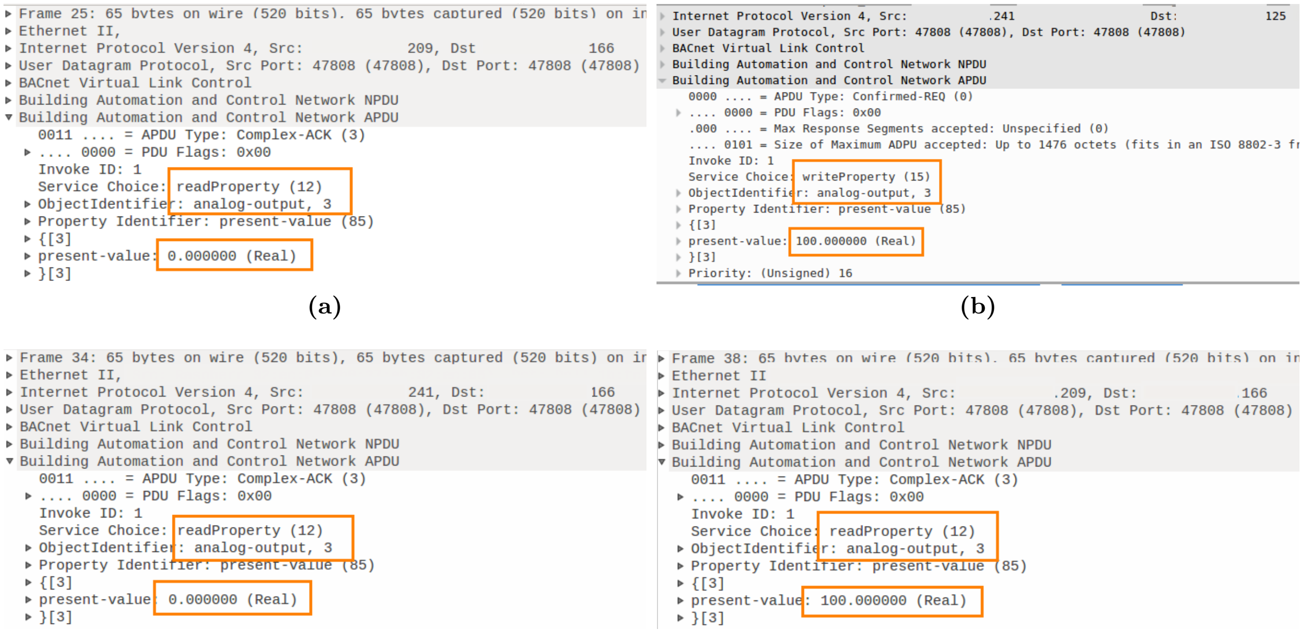
Figure 7: The topology of BAS testbed used in the paper.

First, without cyber-attacks, our emulation shows that the Server is able to receive consistent object values from two paths. Next, we emulated all six attack scenarios where the adversary launches both *false data injection attacks* and *data integrity attacks* from three different locations. We designed similar attack procedures for different attack scenarios, as described below:

1. The Server requests the current value of the interested object *obj* through *ReadProperty* packets, and the both FPs response the value of *obj* as x;
2. The attacker sends a malicious *WriteProperty* request to the PD to change *obj* from value x to value y under three cases: (1) *a compromised FP*, (2) *MITM*, and (3) *attacker inside the network (not MITM)*;
3. The Server sends another *ReadProperty* request to *obj* from two paths. The attacker under Case (1) and (2) manipulates the data response with false sensing value x, trying to hide the malicious changes made on the *obj*; the attacker under Case (3) does not manipulate *ReadProperty* response;
4. The Server receives inconsistent values (under Case (1) and (2)) or unexpected values (under Case (3)), which indicates the network is under attack.

5.2 Detection Performance

Wireshark [24] was used to monitor the network traffic. Fig. 8 presents Wireshark snapshots of attack Case (1): *a single compromised FP*. Specifically, (1) The Server sends a *ReadProperty* request on object *AnalogOutput#3* whose current value is 0.0 (cf. Fig. 8-(a)); (2) The attacker compromises the primary FP and leverages it to send a *WriteProperty* request to the PD to change *AnalogOutput#3* to 100.0 (cf. Fig. 8-(b)); (3) The compromised FP reports false value of *AnalogOutput#3* as 0.0 (cf. Fig. 8-(c)), but the other FP responds with the real value as 100.0 (cf. Fig. 8-(d)); (4) the inconsistent values indicate one FP or path is compromised.



```
▶ Frame 25: 65 bytes on wire (520 bits), 65 bytes captured (520 bits) on ir
▶ Ethernet II,
▶ Internet Protocol Version 4, Src:               209, Dst              166
▶ User Datagram Protocol, Src Port: 47808 (47808), Dst Port: 47808 (47808)
▶ BACnet Virtual Link Control
▶ Building Automation and Control Network NPDU
▼ Building Automation and Control Network APDU
    0011 .... = APDU Type: Complex-ACK (3)
  ▶ .... 0000 = PDU Flags: 0x00
    Invoke ID: 1
    Service Choice: readProperty (12)
  ▶ ObjectIdentifier: analog-output, 3
  ▶ Property Identifier: present-value (85)
  ▶ {[3]
  ▶ present-value: 0.000000 (Real)
  ▶ }[3]
```

(a)

```
▶ Internet Protocol Version 4, Src:           .241                 Dst:          125
▶ User Datagram Protocol, Src Port: 47808 (47808), Dst Port: 47808 (47808)
▶ BACnet Virtual Link Control
▶ Building Automation and Control Network NPDU
▼ Building Automation and Control Network APDU
    0000 .... = APDU Type: Confirmed-REQ (0)
  ▶ .... 0000 = PDU Flags: 0x00
    .000 .... = Max Response Segments accepted: Unspecified (0)
    .... 0101 = Size of Maximum ADPU accepted: Up to 1476 octets (fits in an ISO 8802-3 fr
    Invoke ID: 1
    Service Choice: writeProperty (15)
  ▶ ObjectIdentifier: analog-output, 3
  ▶ Property Identifier: present-value (85)
  ▶ {[3]
  ▶ present-value: 100.000000 (Real)
  ▶ }[3]
  ▶ Priority: (Unsigned) 16
```

(b)

```
▶ Frame 34: 65 bytes on wire (520 bits), 65 bytes captured (520 bits) on ir
▶ Ethernet II,
▶ Internet Protocol Version 4, Src:               241, Dst:             166
▶ User Datagram Protocol, Src Port: 47808 (47808), Dst Port: 47808 (47808)
▶ BACnet Virtual Link Control
▶ Building Automation and Control Network NPDU
▼ Building Automation and Control Network APDU
    0011 .... = APDU Type: Complex-ACK (3)
  ▶ .... 0000 = PDU Flags: 0x00
    Invoke ID: 1
    Service Choice: readProperty (12)
  ▶ ObjectIdentifier: analog-output, 3
  ▶ Property Identifier: present-value (85)
  ▶ {[3]
  ▶ present-value: 0.000000 (Real)
  ▶ }[3]
```

(c)

```
▶ Frame 38: 65 bytes on wire (520 bits), 65 bytes captured (520 bits) on ir
▶ Ethernet II
▶ Internet Protocol Version 4, Src:              .209, Dst:            .166
▶ User Datagram Protocol, Src Port: 47808 (47808), Dst Port: 47808 (47808)
▶ BACnet Virtual Link Control
▶ Building Automation and Control Network NPDU
▼ Building Automation and Control Network APDU
    0011 .... = APDU Type: Complex-ACK (3)
  ▶ .... 0000 = PDU Flags: 0x00
    Invoke ID: 1
    Service Choice: readProperty (12)
  ▶ ObjectIdentifier: analog-output, 3
  ▶ Property Identifier: present-value (85)
  ▶ {[3]
  ▶ present-value: 100.000000 (Real)
  ▶ }[3]
```

(d)

Figure 8: Emulation results of a compromised FP. (a) The Server reads AO#3 as 0.0; (b) The compromised primary FP changes AV#3 to 100.0; (c) The ReadProperty response from the compromised FP; and (d) the ReadProperty response from the good FP.

Table 4 presents that our solution can be used to prevent/detect different attack scenarios. No matter where the attacker launches the attack, when the adversary tries to deceive the Server with false sensor data from one path, our solution can **prevent** such attack because the redundant sensing value(s) can be used to validate the original value. When malicious false command attacks are launched toward Physical Devices, our solution can **detect** such attacks by either directly acquiring control objects/setpoints or acquiring the sensor values which are influenced by the changed control objects in the closed-loop physical dynamics. Our solution could effectively detect malicious and unauthorized attacks and help system operators to quickly take remedies.

Attacks Locations	Data Integrity Attacks	False Command Attacks
Compromised FP	Preventable	Detectable
MITM (Server-FP)	Preventable	Detectable
Attackers inside Network (Not MITM)	Preventable	Detectable

Table 4: Path Redundancy can effectively prevent/detect different attack scenarios.

5.3 Timing Performance

We conducted experiments to measure the latency of data acquisition on an object with Path Redundancy, and compared that with the original solution. We conducted 600 experiments in each case on our testbed. In Path Redundancy case, the Server sends two ReadProperty requests to both FPs who forward the packets to the PD and later forward the corresponding responses back to the Server. The latency of data acquisition consists of the RTT of all ReadProperty messages on the same object as well as the processing and scheduling latency from the devices.

Fig. 9 presents the distribution of data acquisition latency at the Server. The average latency increases from 2.24 ms in the original solution (cf. Fig. 9-(a)) to 3.71 ms in Path Redundancy with one redundant path (cf. Fig. 9-(b)). The additional 1.47 ms timing latency mainly comes from the processing and scheduling latency at the PD. Overall, the additional network latency is minimal (at miliseconds level) so that it will not affect the data acquisition process at the BAS Server.

5.4 Network Traffic Overhead

Our solution only affects the Server-initiated data sensing packets. We measured the network traffic overhead in terms of **Packet Overhead** and **Bandwidth Overhead** based on our previous data collection from a building automation network in a university campus. In BACnet networks, the Server-initiated data sensing services are ReadProperty and ReadPropertyMultiple.

In terms of packet overhead, a single day traffic between the Server and all Field Panels contained 88, 156 packets. 23.56% packets corresponded to the read packets (20.22% ReadPropertyMultiple and 3.34% ReadProperty). When all objects are enabled for Path Redundancy, there will be 19.07% packet overhead for one redundant path and 32.03% packet overhead for two redundant paths.

In terms of bandwidth overhead, a single day traffic contained 519 million bytes. 15.80% bandwidth (82 million bytes) corresponded to the read packets (11.56% ReadPropertyMultiple and 4.24% ReadProperty). When all objects are enabled for Path Redundancy, there will be 13.64% bandwidth overhead for one redundant path and 24.01% bandwidth overhead for two redundant paths.

It is noted that these volumes are low enough that in-

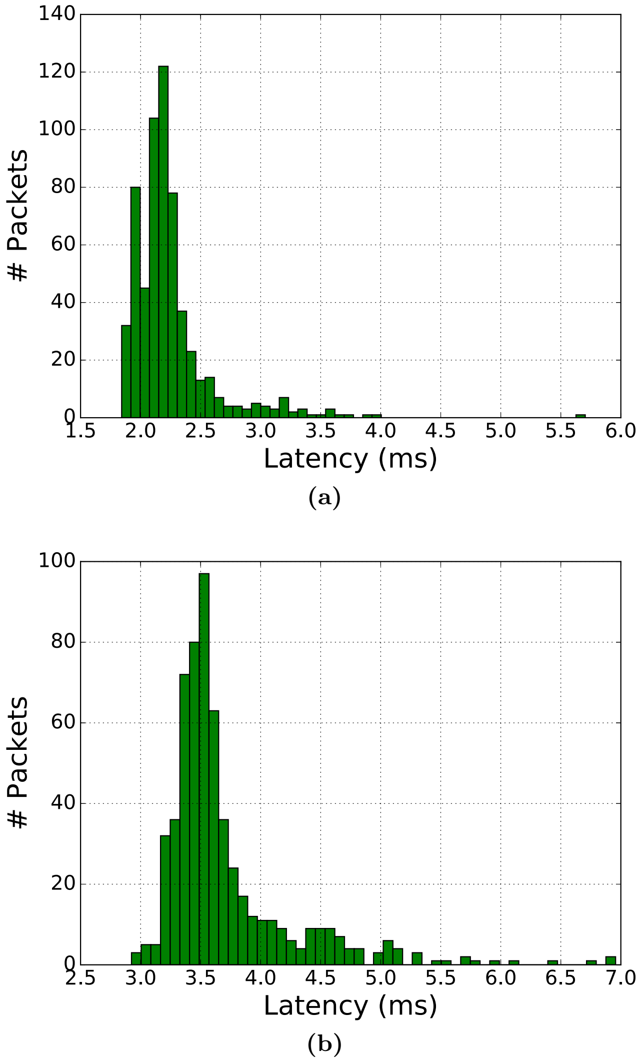


Figure 9: Distribution of data acquisition latency from the BAS Server. (a) data acquisition without our solution; (b) data acquisition with Path Redundancy (one redundant path).

creased bandwidth from Path Redundancy does not cause overload of the network.

6. RELATED WORK

Several works have been conducted to improve the resilience and data integrity of SCADA system through redundancy. Some solutions suggest the use of **redundant hardware** (*e.g.* controllers, substations, PLCs or sensors) to provide redundant data replication [6, 25, 27]; while others focus on different types of **path redundancy** (P2P or pub/sub) to improve data validity [5, 7, 10, 11, 14]. Since path redundancy does not require additional hardware devices, it is more cost-efficient and can be easily integrated with the current SCADA deployment practices. Different from our approach, previous path redundancy work focuses on the path diversity in the Server-PLC channels and thus is not able to detect a compromised PLC.

6.1 Redundant Hardware

The redundancies at hardware level and data replicas provide desired operation to CPS with failed nodes. Zhang et al. [27] present a general framework that abstracts the essential properties of sensor networks for the identification of compromised sensor nodes. They assume the network consists of a large number of redundant sensor nodes. Their application-independent approach does not need to consider dynamics of physical systems.

The authors in [25] describe the main deficiencies in the current communication networks of power grid and propose a new two-fold information architecture with various redundancy configurations. It adopts suitable computing and communication techniques to take into account the requirements of real-time data, security, availability, scalability, and appropriate Quality of Service (QoS). Multi-protocol label switching (MPLS), Virtual Private Networks (VPN), and firewalls are applied to meet security requirement. This solution requires significant changes to the power grid in both hardware and software.

Cai et al. [6] develop a reliable remote control system for subsea blowout preventer stack. The remote control system is based on the off-the-shelf triple modular redundancy system and various redundancy techniques such as controller redundancy, bus redundancy and network redundancy. They show that when faults happen on PLCs, discrete input groups and analog input groups, alarms will be raised at the HMI.

6.2 Path Redundancy

The authors in [5] propose an agent-based layer on top of a P2P (peer-to-peer) network to improve message exchange reliability. It also discusses the advantages and disadvantages of using Chord network [20] for power grid applications.

Germanus et al. [10, 14] propose a P2P overlay network to increase the resilience of SCADA systems. The P2P communication between a MTU (Master Terminal Units) and multiple RTUs (Remote Terminal Units) provides path redundancy and data replication. A middleware is inserted between the SCADA application and the IP layer at every SCADA device. The middleware extracts SCADA payload and stores the data in the P2P overlay. When a MTU receives a message from SCADA network, it requests the same message from different replicas in the P2P network to validate the original message. Their approach can protect SCADA from node crashes and data integrity attacks that are located between the source and destination. However, they cannot detect malicious RTUs colluding on behalf of an adversary. Their intrusive middleware-based approach is scalable to accommodate ongoing developments of interconnected SCADA systems.

In contrast, our approach is shown to be able to detect a compromised PLC/RTU which may be used by an adversary to launch data integrity attacks and false command attacks. Compared with their intrusive middleware-based approach, our solution takes advantage of the existing PLCs and equipment in the system and thus can be more easily applied to current deployment practices.

Germanus et al. [9] further proposed Coral, a decentralized P2P protocol that provides low latency and reliable convergecast for sensor data collection. Coral consists of three parts: (1) periodic link latency measurements, (2) latency minimal and disjoint path search, and (3) robust and reliable convergecast routing. Their protocol is evaluated using a case study: system protection workload on a realistic power transmission network topology.

The authors in [11] propose a publish-subscribe middleware framework to meet the data delivery and surveillance requirements for electrical power grid. Data streams produced at the source (publisher) are distributed to destinations (subscribers) without the publisher having to track all the subscribing entities. The pub/sub structure requires message broker nodes to manage forwarding and receiving mechanism. This framework is suitable for a sensing-only network where multiple devices are interested in the same data from the same device. It explores the path diversity between the publisher and its subscriber(s), rather than the path between the physical devices and the publisher. Thus, they are not able to detect/prevent attacks from a compromised subscriber/path. The paper evaluated the performance in terms of forwarding latency and load scalability, but failed to provide evaluation on security performance or redundant paths.

7. CONCLUSION

Path diversity has been explored in many contexts before. This paper shows that Path Redundancy is a simple and effective solution to counter compromised PLCs in CPS. The proposed redundant paths are enabled with existing PLCs and are only used for sensing purposes. We considered six attack scenarios and described a demonstration of our techniques in an emulated Building Automation System using Raspberry Pi devices and a PC-based control server. Emulation results have shown that our solution could effectively prevent data integrity attacks and detect false command attacks from a single source. Our solution is cost-efficient, easy-to-deploy, and suitable for different types of CPS with different communication interfaces. In future work, we plan to migrate the control of a physical device from a single designated PLC to other possible PLCs. We also plan to use virtualized PLCs for both sensing and actuation purposes. Such design will allow us to convert current architecture from tree structure to general switched structure.

8. ACKNOWLEDGEMENTS

This research is supported in part by the National Science Foundation under Grant no. CNS 1218929 and the Qatar National Research Foundation 9th Cycle under Grant no. 9-069-1-018. We would like to thank Carl Neilson and David Fisher who provided insight and expertise to help us understand the problem.

9. REFERENCES

[1] 1815-2012 - IEEE Standard for Electric Power Systems Communications-Distributed Network Protocol (DNP3). *IEEE Std. 1815-2012*, 2012.

[2] Z. Alliance et al. Zigbee specification, 2006.

[3] S. ASHRAE. Standard 135-1995: Bacnet-a data communication protocol for building automation and control networks. *American Society of Heating, Refrigerating and Air-Conditioning Engineers, Atlanta, Georgia, USA*, 1995.

[4] R. R. R. Barbosa. Anomaly detection in scada systems-a network based approach. *PhD thesis*, 2014.

[5] H. Beitollahi and G. Deconinck. Analyzing the chord peer-to-peer network for power grid applications. In *Fourth IEEE Young Researchers Symposium in Electrical Power Engineering*, page 5, 2008.

[6] B. Cai, Y. Liu, Z. Liu, F. Wang, X. Tian, and Y. Zhang. Development of an automatic subsea blowout preventer stack control system using plc based scada. *ISA transactions*, 51(1):198–207, 2012.

[7] G. Deconinck, T. Rigole, H. Beitollahi, R. Duan, B. Nauwelaers, E. Van Lil, J. Driesen, R. Belmans, and G. Dondossola. Robust overlay networks for microgrid control systems. In *Proc. Workshop on Architecting Dependable Systems (WADS '07)*, pages 148–153, 2007.

[8] B. Electronics. Rs-485 tips, tricks, questions answers. 2016.

[9] D. Germanus, A. Khelil, J. Schwandke, and N. Suri. Coral: Reliable and low-latency p2p convergecast for critical sensor data collection. In *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 300–305. IEEE, 2013.

[10] D. Germanus, A. Khelil, and N. Suri. Increasing the resilience of critical scada systems using peer-to-peer overlays. In *Architecting Critical Systems*, pages 161–178. Springer, 2010.

[11] H. Gjermundrod, D. E. Bakken, C. H. Hauser, and A. Bose. Gridstat: A flexible qos-managed data dissemination framework for the power grid. *IEEE Transactions on Power Delivery*, 24(1):136–143, 2009.

[12] M. E. Hazen. Understanding some basic recommended standards for serial data communications-a comparison of rs-232, rs-422 and rs-485. *http://www. intersil. com/data/wp/WP0585. pdf*, 2003.

[13] S. Karg. Bacnet stack: An open source bacnet protocol stack for embedded systems, 2015. [Online; accessed 17-November-2016].

[14] A. Khelil, D. Germanus, and N. Suri. Protection of scada communication channels. In *Critical Infrastructure Protection*, pages 177–196. Springer, 2012.

[15] R. Langner. To kill a centrifuge: A technical analysis of what stuxnet's creators tried to achieve. *The Langner Group*, 2013.

[16] R. M. Lee, M. J. Assante, and T. Conway. Analysis of the cyber attack on the ukrainian power grid. *SANS Industrial Control Systems*, 2016.

[17] I. Modbus Organization. Modbus home page. http://www.modbus.org/.

[18] NCCIC/ICS-CERT. Alert (ir-alert-h-16-056-01): Cyber-attack against ukrainian critical infrastructure. 2016. [Online; accessed 25-November-2016].

[19] Z. Pan, S. Hariri, and Y. Al-Nashif. Anomaly based intrusion detection for building automation and control networks. In *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, pages 72–77. IEEE, 2014.

[20] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review*, 31(4):149–160, 2001.

[21] A. Valdes and S. Cheung. Communication pattern anomaly detection in process control systems. In *IEEE Conference on Technologies for Homeland Security (HST '09)*, pages 22–29. IEEE, 2009.

[22] Wikipedia. Rs-485 — wikipedia, the free encyclopedia, 2016. [Online; accessed 17-November-2016].

[23] Wikipedia. Stuxnet — wikipedia, the free encyclopedia, 2016. [Online; accessed 17-November-2016].

[24] Wireshark. https://www.wireshark.org/.

[25] Z. Xie, G. Manimaran, V. Vittal, A. Phadke, and V. Centeno. An information architecture for future power systems and its reliability analysis. *IEEE Transactions on Power Systems*, 17(3):857–863, 2002.

[26] D. Yang, A. Usynin, and J. W. Hines. Anomaly-based intrusion detection for scada systems. In *NPIC&HMIT'05*, pages 12–16. Citeseer, 2006.

[27] Q. Zhang, T. Yu, and P. Ning. A framework for identifying compromised nodes in sensor networks. In *Securecomm and Workshops, 2006*, pages 1–10. IEEE, 2006.

[28] B. Zhu and S. Sastry. Scada-specific intrusion detection/prevention systems: a survey and taxonomy. In *Proceedings of the 1st Workshop on Secure Control Systems (SCS '10)*, 2010.